Proceedings Paper:
Dale, Matthew Nicholas, Miller, Julian Francis orcid.org/0000-0002-7692-9655, Stepney, Susan orcid.org/0000-0003-3146-5401 et al. (1 more author) (2016) Evolving Carbon Nanotube Reservoir Computers. In: Amos, Martyn and Condon, Anne, (eds.) Unconventional Computation and Natural Computation:15th International Conference, UCNC 2016, Manchester, UK, July 11-15, 2016, Proceedings. 15th International Conference Unconventional Computation and Natural Computation (UCNC), 15 Aug 2016 Lecture Notes in Computer Science . Springer , GBR , pp. 49-61.

https://doi.org/10.1007/978-3-319-41312-9_5

Evolving Carbon Nanotube Reservoir Computers

Matthew Dale[1,3], Julian F. Miller[2,3], Susan Stepney[1,3], Martin A. Trefzer[2,3]

[1]Department of Computer Science, University of York, UK
[2]Department of Electronics, University of York, UK
[3]York Centre for Complex Systems Analysis
md596@york.ac.uk

Abstract. Reservoir Computing is a useful general theoretical model for many dynamical systems. Here we show the first steps to applying the reservoir model as a simple computational layer to extract exploitable information from physical substrates consisting of single-walled carbon nanotubes and polymer mixtures. We argue that many physical substrates can be represented and configured into working reservoirs given some *pre-training* through evolutionary selected input-output mappings and targeted input stimuli.

Keywords: Material Computation, Evolution-*in-Materio*, Reservoir Computing, Unconventional Computing, Evolvable Hardware

1 Introduction

Reservoir Computing (RC) [6, 11] has been proposed as an expressive model and as a computationally inexpensive method for training rich high-dimensional dynamical systems, ranging from simulated and biological neural networks to novel hardware-based implementations [9].RC exploits the emergent complexity of dynamic networks to perform information processing tasks.

An input-driven Reservoir Computer is typically divided into three parts: the input, the "reservoir", and the readout. This separation provides a representation that exploits the complex projection of the input into a high-dimensional state space. This rich state space is created from a *black-box* network and is harnessed using only a simple output training mechanism.

Most reservoirs are hand-crafted to a task, so there is often a need for expert domain knowledge to design an optimal system. However, due to the system partitioning, some element of semi-autonomous *pre-training* is possible, avoiding the need for manual search for efficient reservoirs. This pre-training concept appears in the RC literature [9], but is typically used only for simulated reservoirs. We hypothesise that pre-training can be highly advantageous when moving into the physical domain.

Evolution-in-Materio (EIM) [12, 13] explores the concept of *configuring* matter for computation, originally outside the context of RC. The training procedure uses an evolutionary algorithm to configure a rich continuous complex material

to perform desired tasks. This usually takes the form of evolving a set of signals or static voltages and their connection locations on an electrode array interfacing the computational material. The aim is to evolve an input-output mapping that carries out a desired computational mapping. Its rationale is that physical systems contain enormous amounts of complexity, and that evolution exhibits the most efficient method to discover and exploit these physical properties.

Here we investigate the use of computer controlled evolution (CCE) to configure a physical system for RC. We demonstrate that by using a form of evolution-in-materio we can pre-train a physical dynamical system – which might not necessarily be a natural reservoir candidate – into a functional and optimisable reservoir computing system. We demonstrate this on two temporal reservoir computing tasks: the Nonlinear Auto Regressive Moving Average task (NARMA) and the wave generator task, each requiring different internal characteristics. We compare four different carbon nanotube-based materials, a conductive sheet, and an open-circuit system.

2 Reservoir Computing

Reservoir computing exploits the dynamic response of an excitable system given a single- or multi-dimensional input signal. Typically, the reservoir has some nonlinear properties, enabling both dynamic memory and dynamic processing. RC has become a competitive technique for training Recurrent Neural Networks (RNNs) on temporal processing tasks.

The conceptual view of what makes a "reservoir" and the methods used to train them is not limited to simulated neural networks. For example, Optoelectronic and Photonic [1, 15, 19] reservoir-based systems can be made. Such highly specialised reservoirs require some amount of *pre-* and *post*-processing.

However, other material reservoirs require minimal additional processing. For example, one of the first *physical* reservoirs was simply a bucket of water [5]. A fabricated neuromorphic device called an Atomic Switch Network (ASN) has been modelled as a reservoir [18, 17].

There, communication with the reservoir—a configurable memristive network—takes place through a multiple input/output micro-electrode array. The memristive substrate contains a random topology of highly-integrated functionalised silver nanowires that together create emergent behaviours.

We use a basic model reservoir (based on the Echo State Network [6]) consisting of a randomly initialised recurrent *tanh*-based neural network with n nodes. The input(s) to the network $u(n)$ are fed through connection weights W^{in} with a one-to-one mapping to internal nodes. The internal nodes are mapped to each other via the random W matrix, creating the recurrent structure through internal loops. The output of the system $y(n)$ is given by the matrix multiplication of trainable output weights W^{out} and the reservoir's internal states $x(n)$.

We consider *supervised* machine learning tasks where both the training input $u(n)$ and target output $y^{Target}(n)$ are provided. Training is carried out by adjusting W^{out} to reduce the error between the system output $y(n)$ and target

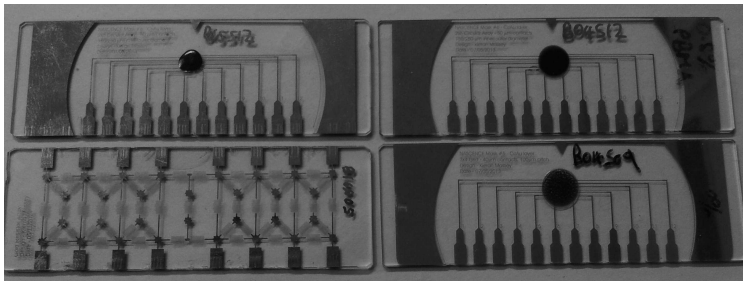

Fig. 1: Substrates under test. Top left, SWCNT/PBMA mixture with a concentration of 1% SWCNT by weight. Top right, SWCNT/PBMA 0.53%. Bottom left, gold resistor array. Bottom right, SWCNT/PMMA 0.1%

output $y^{Target}(n)$. To evaluate the reservoir's performance we partition the data sets into three parts: the *training* set (50%), the *validation* set (25%), and the *test* set (25%). The output weights are trained on the training set, and reservoir fitness is evaluated on the validation set. The final error is calculated using the *Normalised Root Mean Squared Error*(NRMSE) on the test set data.

2.1 Optimising Reservoirs

Simulated Echo State Reservoirs have many parameters. One can change their dynamics and memory capacity by adjusting the global scaling factors for the weights. For example, the *spectral radius* ρ, a scaling parameter for the internal weights W, can dramatically influence the *echo state property* [6] (fading memory capacity) of the system. Other parameters such as topology, neuron sparsity and type of activation function can also be varied. This suggests that a certain amount of optimisation can be done over a "randomly" created reservoir.

A number of optimisation techniques have been explored in simulated networks, but few, if any, have been used in hardware-based reservoirs. This raises the question: can we create and train systems that might not 'naturally' form reservoirs, or are in their untrained state classed as poor reservoirs? We hypothesise that a system with interesting and malleable properties can be configured, or dynamically perturbed, into a state that produces effective reservoir properties.

3 Materials and Hardware

3.1 Materials under investigation

The materials used here were fabricated within the NASCENCE consortium [4]. The aim of that project was to investigate candidate materials and techniques for configuring materials for computation. Our work continues this agenda by investigating a substrate's response to reservoir-style training, to demonstrate that the reservoir computing model can be applied to a range of substrates.

Our experimental method is evaluated on 4 different material test subjects (fig. 1) and two additional system "settings" (*short-circuit* and *open-circuit*) to provide both a description of how much the system as a whole is being evolved and to show what can be achieved with a purely conductive sheet.

The material for each test subject is deposited onto a glass slide with 12 chromium/gold-contact (40 to 50 µm contact diameter and 100 to 150 µm contact spacing) micro-electrodes arranged in either a circle or square array.

Test subjects one and two are single-wall Carbon Nanotube (SWCNT)/ polymer mixtures with SWCNT concentrations of 0.53% and 1% (by weight) mixed with poly-butyl-methacrylate (PBMA) dissolved in Anisole. Test subject three is a 0.1% SWCNT mixture with poly-methyl methacrylate (PMMA). For each substrate, approximately 20ml of the mixture is dispensed on the electrode array, then dried. The random formations and settling of SWCNTs within the samples can fluctuate. The conductivity of each material is determined by SWCNT density and electrode contact. The heterogeneous behaviour of the material is the result of the dielectric properties of the polymer and the shifting electronic properties of networks formed from both semi-conducting and metallic SWCNTs.

Test subject four is a reference material: a gold resistor array patterned onto a glass slide with multiple connection points using etch-back photo-lithography. The resistor array is arguably simpler and reasonably stable with known internal resistance values. This test subject investigates if the technique can be applied to more linear mediums and what, if any, are the advantages of SWCNT-based materials over simple resistive networks.

The *open* and *short* circuit settings are added to verify the significance the material has on the evolvability of the system, that is, to pinpoint what is doing the computation. In the open-circuit no material is connected; the system is simply left to find a solution through system noise, or from unknown characteristics within the system. The conductive sheet (copper tape) is used as a short-circuit connection to assess if the material has any advantageous properties beyond conductivity.

3.2 Hardware platform

The hardware used in this experiment forms a hybrid digital/analogue hardware loop. Computer controlled evolution (CCE) is performed in the digital space on a connected desktop PC using a MATLAB interface. In the analogue/physical space, the material is stimulated using a National Instruments Data Acquisition Card (NI PCI-6723) supplying analogue output signals, which can be routed to any of the electrodes interfacing the material via an Analog Devices (AD75019) 16×16 analogue cross-point switch. An NI DAQ card (NI PCI-6225) is used to record analogue inputs from the electrode array via the cross-point switch in the same manner.

The cross-point switch is used to autonomously assign which electrodes and DAQ card channels are currently in use and what role each electrode performs. Once the evolved configuration is registered on the cross-point switch, bidirectional communication is established between both DAQ cards and the electrodes.

4 Material Configuration

As part of the NASCENCE project a number of stimulation signals have been investigated, such as complex signals like evolved square waves [10, 14]. For the purpose of this investigation we are restricting ourselves to static voltages to avoid any interference, or artefacts, that evolution may create in respect to temporal-based tasks.

The electrical configuration of a material is therefore exclusively carried out through the placement and adjustment of static voltages. The aim is to configure the internal characteristics of the material by manipulating its natural dynamics, conductivity and signal processing abilities.

To encode the electrical configuration of the substrate a 21-gene genotype is created. All genes are open to mutation and are subdivided into: electrode assignment (genes 1–12), redundant genes (genes 13–16), values of static input voltages (genes 17–20) and input scaling on $u(n)$ (gene 21). Genes 1–16 are integer values; all other genes are floating point numbers with a precision of 4 decimal places; genes 17–20 range between [−5V,5V]; gene 21's range is [0V,2V] for the NARMA task (already pre-scaled by factor of 10) and [0V,5V] for the wave generator task.

The phenotype of the system is implemented via the cross-point switch assignment. The interfacing equipment is set up so that all accessible inputs and outputs are connected to the switch. The switch then directs which DAQ channels communicate to the electrode array via a 256-bit digital input (SIN) derived from the values in genes 1–12.

This genome design allows evolution to decide both the number of readouts in use and the number of static input voltages the material can receive. At genotype instantiation, and under the mutation operator, a maximum of 10 possible readouts (referred to as measurable *reservoir states*) are possible.

This is due to the input signal $u(n)$ and ground (GND) always being required. A maximum of 4 static input voltages (referred to as "configuration voltages") can also be applied simultaneously. This feature (implemented by redundant genes) allows evolution to converge towards any assignment, such as 6 readouts and 4 configuration voltages, or, 8 readouts and 2 configuration voltages (fig. 2), as long as the required phenotype size of 12 is always adhered to.

The input scaling gene (scaling $u(n)$) is added as the material may require varying input-data intensities under different electrical configurations. The gene is initialised at the maximum value, then left to evolve.

We use an elitist $1+\lambda$ evolutionary strategy with a population of 5 ($\lambda = 4$) for 150 generations across 10 runs. The λ children are mutations of the previous generation's fittest individual. In the case that a child is the same fitness as the parent, the child is selected to pass on its genes. This allows evolution to neutrally sweep the search space if no immediate fitness change is present.

The procedure is shown in fig. 3. First, a material is selected, equating to a random initialisation of a simulated network. Next, the evolutionary run commences, cycling through the generational loop for every new population. This loop comprises: a physical resetting (grounding) of the material; the application

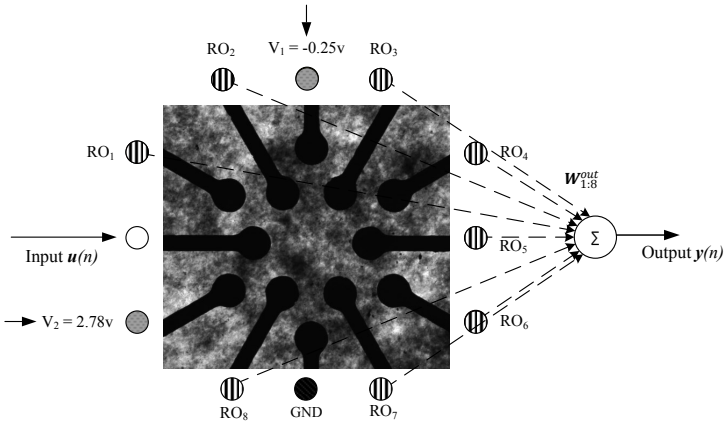

Fig. 2: Physical reservoir representation using electrodes. Each assigned readout electrode (RO_n) forms the reservoir state $x_n(n)$. The configuration voltages (V_n) location and value are decided upon by evolution. The W^{out} matrix is calculated and applied in the digital domain.

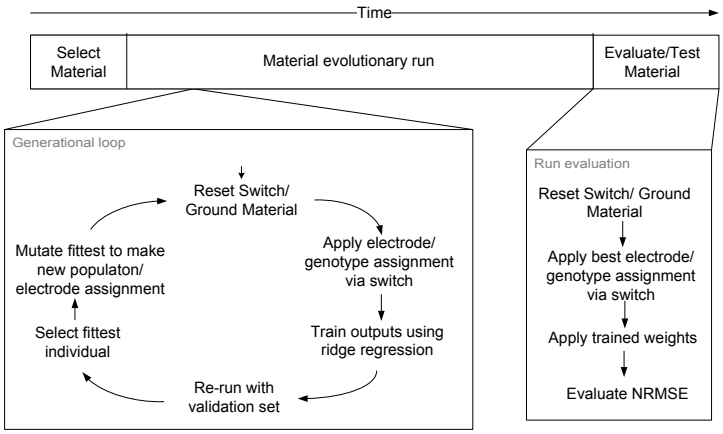

Fig. 3: Reservoir work flow through time: the combined evolutionary-regression training procedure for the hardware-based reservoir. The generational loop is expanded to show the switch assignment process and training/validation process. The final evaluation procedure using the test set is also expanded.

of a new switch assignment (material "configuration") from the genotype for every individual; and a ridge regression (using Tikhonov regularisation) training step on the electrode output weights. The fitness of each individual in the generational loop is calculated on the validation set using NRMSE. The result, calculated on the best individual found in the evolutionary run, is the error calculated on the "unseen" test set.

5 Benchmark tasks

5.1 Nonlinear Auto-Regressive Moving Average (NARMA) task

The NARMA task originates from work on training recurrent networks [2]. It evaluates a reservoir's ability to model an n-th order highly non-linear dynamical system where the system state depends on the driving input as well as its own history. The challenging aspect of the NARMA task is that it contains both non-linearity and long-term dependencies created by the n-th order time-lag.

An n-th ordered NARMA experiment is carried out by predicting the output $y(n+1)$ given by eq.(1) when supplied with $u(n)$ from a uniform distribution of interval [0, 0.5]. For the 5-th and 10-th order systems $\alpha = 0.3$, $\beta = 0.05$, $\delta = 10$ and $\gamma = 0.1$.

$$y(n+1) = \alpha y(n) + \beta y(n)\left(\sum_{i=0}^{\delta} y(n-i)\right) + 1.5u(n-\delta)u(n) + \gamma \qquad (1)$$

5.2 Wave Generator task

The wave generator task requires a rich transformation of an input waveform (a periodic signal) to create a new waveform using temporal features such as phase shifts, delays, harmonic generation, recurrence etc. The task [17] is linked directly to Fourier series analysis. The task is to train a reservoir to produce three different output waveforms given an input sine wave. This is achieved by applying an input sine-wave to one electrode, to produce a square-wave, sawtooth, and cosine waveform of the same frequency, and a sine-wave with double frequency at $y(n)$.

5.3 Memory Capacity

Measuring the short-term memory capacity of a reservoir was first outlined in [7] as a quantitative measurement of the echo state property (fading memory). To determine the memory capacity of a reservoir we measure how many delayed versions of the input $u(n-k)$ the outputs can recall or recover with precision. Applying eq. (2), we can measure memory capacity by how much variance of the delayed input can be recovered, summed over all delays. This is carried out by training individual output units to recall the input at time k.

$$MC = \sum_{k=1}^{\infty} MC_k = \sum_{k=1}^{\infty} \frac{cov^2(u(n-k), y(n))}{\sigma^2(u(n))\sigma^2(y(n))} \qquad (2)$$

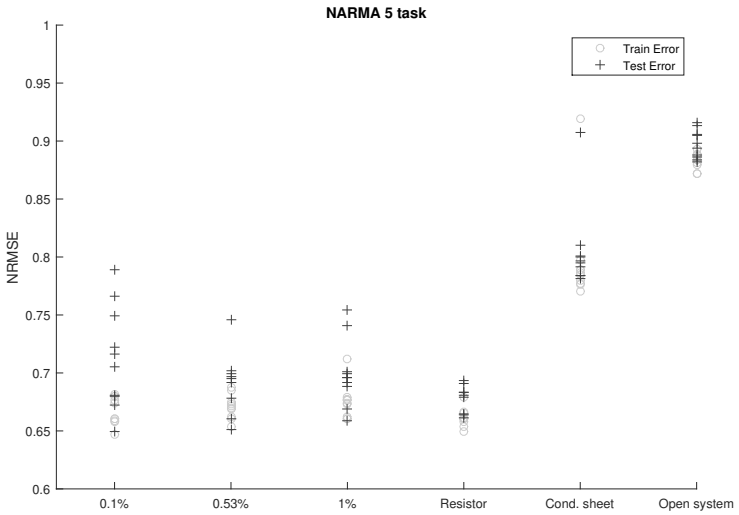

Fig. 4: NARMA-5 plot of train and test error of 10 runs across all materials (lower error is better fitness). All test subjects outperform the open system; all with the exception of some 0.1% concentration runs outperform the conductive sheet in all runs.

6 Experimental Results

Fig.4 shows the NARMA-5 task results. The materials under test, when using the evolved configurations, outperform both the conductive sheet and the open system. This suggests the material is a significant element to the overall computational system, an assumption made within the literature but formally verified here, and that the computational properties of the material are trainable through evolution.

The resistor array produces better results with a smaller variance than the SWCNT materials on this task. However, this almost reverses for the test error when moving to the harder NARMA-10 task.

Fig.5 shows the NARMA-10 task results. The materials perform modestly on this task given the availability of trainable states (readouts). Although an exact comparison cannot be made, some indication of system performance on this task can be seen by looking at an optoelectronic reservoir [15] consisting of a 50-node psuedo-network reaching an NRMSE $\approx$ 0.41, and various sized simulated-reservoirs ranging from an NRMSE of 0.4 to 0.9 in [20].

The required memory capacity (MC) for each task correlates to the input lag and is therefore different for the two NARMA tasks. The measured MC does not change, however (fig.6). It could be that the material cannot increase its MC, given the small number of readouts available. Nevertheless it is puzzling: the MC should not be limited by the number of readouts, because the internal structure and dynamics of the system do not possess the same limitations. Al-

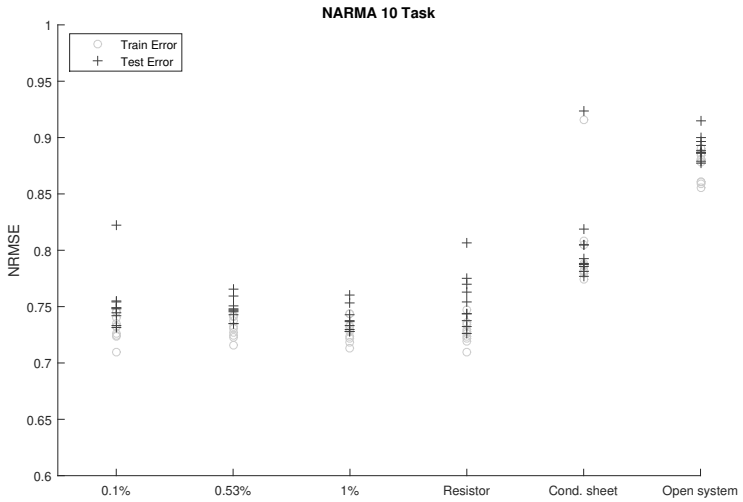

Fig. 5: NARMA-10 plot of train and test error of 10 runs across each material. Despite an increase in complexity the material still shows some computational advantage. The resistor array appears to struggle more on generalisation of the test data on this task.

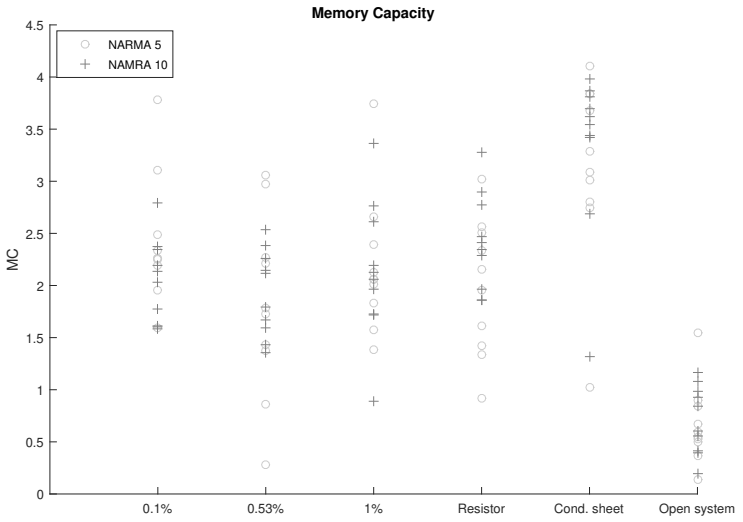

Fig. 6: Memory Capacity of all test subjects post-evolutionary configuration, i.e. evaluated on the best configuration found from each run.

Material	Saw(best/avg)	Cos(best/avg)	Square(best/avg)	2Sin(best/avg)
PMMA (0.1%)	0.347/0.487	0.058/0.079	0.266/0.293	**0.242**/0.787
PBMA (0.53%)	**0.325**/4.358	**0.015**/2.915	0.289/2.074	0.255/8.986
PBMA (1%)	0.417/0.569	0.029/0.069	**0.253**/0.308	0.348/0.881
Resistor array	0.375/0.499	0.031/0.037	0.261/0.382	0.262/0.705
Cond. sheet	0.482/3.262	0.374/1.025	0.367/3.619	0.669/0.895
Open system	0.697/0.750	0.102/0.121	0.579/0.669	0.999/1.000
ASN (MSE)	0.1071	0.0028	0.0451	0.0910
PBMA (0.53%)	0.0352	0.0001	0.0830	0.0325

Fig. 7: Wave Generator results for an input 1 kHz sine wave. Test Error is given for 10 runs. Additional MSE results are added for the PBMA (0.53%) against ASN measurements given in [18].

ternatively, the method used to evaluate the material's memory capacity could be too susceptible to noise.

The open system has a very small MC in comparison. The conductive sheet, however, appears on average to possess a consistently larger MC. All of our test subjects appear to fit within this range, making it somewhat difficult to determine significant behavioural differences between the test subjects using memory capacity alone.

For the wave generator task an input frequency of 1 kHz was used, rather than the 10 Hz chosen in [18], as there is evidence that SWCNT/polymer materials produce more interesting behaviours at higher frequencies [16]. Results are shown in fig.7. The PBMA (0.53%) material shows the best configuration averaged across all waveforms; however, across the 10 runs more poor solutions are found compared to the other materials. Fig. 8 shows the trained outputs of the configured PBMA (0.53%) material for each waveform; visually we can see a variation in performance across the waveforms, and in particular, the increased difficulty experienced on the sawtooth task.

From our results we see that the test materials possess a variety of exploitable electrical properties that may not naturally occur without targeted stimulation. Fig. 9 highlights this by showing an increase in harmonic behaviour that occurs only under configuration: the sub-plot shows only the first three harmonics occur when unconfigured, versus eight or more harmonics when configured.

7 Discussion and Further Work

We have demonstrated that we can evolve configurations that make certain substrates into trainable computational reservoirs. We have demonstrated that small, configurable (analogue) devices can be trained to tackle difficult system modelling and temporal tasks. The results provide an insight into the potential of the methodology, which is not limited to carbon-nanotube based materials.

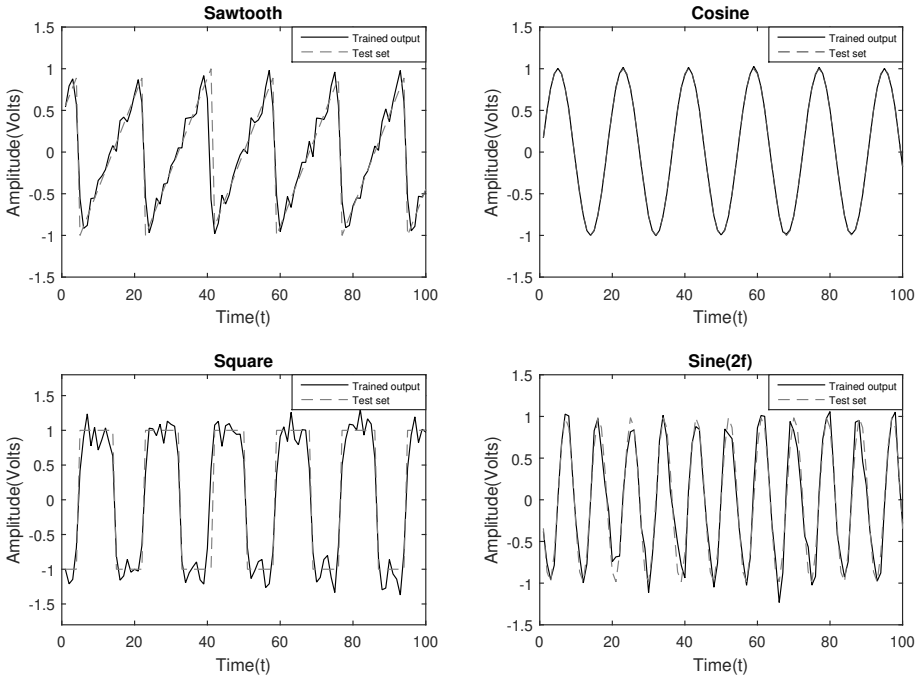

Fig. 8: Trained reservoir output (PBMA 0.53%) plotted against the desired output for each waveform. All waveforms are trained and outputted simultaneously.

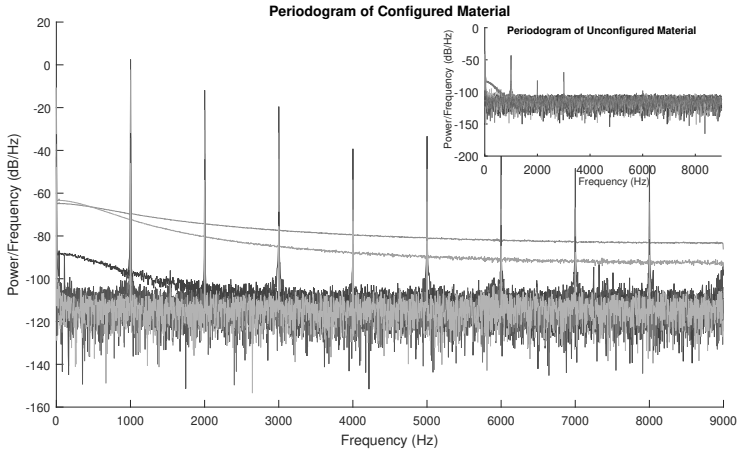

Fig. 9: Power Spectral Density of readout states on the wave generator task for the PBMA 0.53% material. The main plot shows an increase in harmonic behaviour when the material is "configured". The subplot shows the unmodified material given an input 1 kHz sine-wave.

However, these results do leave room for improvement. Our results for the NARMA-10 task are modest in comparison to an optoelectronic reservoir (as shown in §6). However, the latter uses a much larger reservoir (50 nodes) and a different reservoir encoding: representation through a pseudo network using pre/post-processing and a long delay line. For the wave generator task, our configured materials are competitive with or outperform the Atomic Switch Network in [17].

The biggest limitation is the size of our current electrode array. The 6–10 input electrodes (and hence reservoir nodes in the model) is small in comparison to typical numbers of nodes in simulated and hardware-based reservoirs (hundreds). For larger electrode arrays we predict an increase in performance, as the training procedure should have an increased number of internal states and spatial diversity to exploit. We are currently increasing the array size to 64 electrodes, which requires hardware upgrades. This will allow us to undertake more complex temporal tasks.

Other fundamental investigations are still required, such as correlating electrical characteristics to evolved solutions and estimating the information processing capabilities of each material by examining its reservoir *quality*. To do the latter, we will exploit a number of proposed metrics [3, 8, 9]. This involves the quantitative measurement of dynamics and associated reservoir properties. This should provide an improved understanding of how useful these materials are, and how they might behave over a wider range of computational problems.

This work is a first step towards a method in which substrates can be manipulated, or exploited, to extract machine-learning capabilities from an inherently analogue (physical) medium. This can offset the computational load on, or remove the requirement for, digital signal processing for certain tasks. Potential tasks include: collecting and processing sensor data, implementing feature extraction, filtering, controlling a physical system such as a robot.

Acknowledgments

This work was funded by a Defence Science and Technology Laboratory (DSTL) PhD studentship.

The authors thank the EU NASCENCE Project (http://www.nascence.eu) for providing the SWCNT materials used in this work.

References

1. L. Appeltant, M. C. Soriano, G. Van der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. R. Mirasso, and I. Fischer. Information processing using a single dynamical node as complex system. *Nature Communications*, 2:468, 2011.
2. A. F. Atiya and A. G. Parlos. New results on recurrent network training: unifying the algorithms and accelerating convergence. *IEEE Transactions on Neural Networks*, 11(3):697–709, 2000.

3. N. Bertschinger and T. Natschläger. Real-time computation at the edge of chaos in recurrent neural networks. *Neural Computation*, 16(7):1413–1436, 2004.
4. H. Broersma, F. Gomez, J. Miller, M. Petty, and G. Tufte. Nascence project: Nanoscale engineering for novel computation using evolution. *International Journal of Unconventional Computing*, 8(4):313–317, 2012.
5. C. Fernando and S. Sojakka. Pattern recognition in a bucket. In *Advances in Artificial Life*, pages 588–597. Springer, 2003.
6. H. Jaeger. The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148:34, 2001.
7. H. Jaeger. *Short term memory in echo state networks*. GMD-Forschungszentrum Informationstechnik, 2001.
8. R. Legenstein and W. Maass. What makes a dynamical system computationally powerful. *New directions in statistical signal processing: From systems to brain*, pages 127–154, 2007.
9. M. Lukoševičius and H. Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009.
10. Odd Rune Lykkebo and Gunnar Tufte. Comparison and evaluation of signal representations for a carbon nanotube computational device. In *IEEE International Conference on Evolvable Systems (ICES 2014)*, pages 54–60. IEEE, 2014.
11. W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560, 2002.
12. J. F. Miller and K. Downing. Evolution in materio: Looking beyond the silicon box. In *NASA/DoD Conference on Evolvable Hardware 2002*, pages 167–176. IEEE, 2002.
13. J. F. Miller, S. Harding, and G. Tufte. Evolution-in-materio: evolving computation in materials. *Evolutionary Intelligence*, 7(1):49–67, 2014.
14. Stefano Nichele, Odd Rune Lykkebo, and Gunnar Tufte. An investigation of underlying physical properties exploited by evolution in nanotubes materials. In *2015 IEEE Symposium Series on Computational Intelligence*, pages 1220–1228. IEEE, 2015.
15. Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar. Optoelectronic reservoir computing. *Scientific Reports*, 2, 2012.
16. O. R. Lykkeb S. Nichele, D. Laketic and G. Tufte. Is there chaos in blobs of carbon nanotubes used to perform computation? In *Future Computing 2015, The Seventh International Conference on Future Computational Technologies and Applications*, pages 12–17, 2015.
17. H. O. Sillin, R. Aguilera, H. Shieh, A. V. Avizienis, M. Aono, A. Z. Stieg, and J. K. Gimzewski. A theoretical and experimental study of neuromorphic atomic switch networks for reservoir computing. *Nanotechnology*, 24(38):384004, 2013.
18. A. Z. Stieg, A. V. Avizienis, H. O. Sillin, R. Aguilera, H. Shieh, C. Martin-Olmos, E. J. Sandouk, M. Aono, and J. K. Gimzewski. Self-organization and emergence of dynamical structures in neuromorphic atomic switch networks. In *Memristor Networks*, pages 173–209. Springer, 2014.
19. K. Vandoorne, P. Mechet, T. Van Vaerenbergh, M. Fiers, G. Morthier, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman. Experimental demonstration of reservoir computing on a silicon photonics chip. *Nature Communications*, 5:3541, 2014.
20. D. Verstraeten, B. Schrauwen, M. D'Haene, and D. Stroobandt. An experimental unification of reservoir computing methods. *Neural Networks*, 20(3):391–403, 2007.